

Algorithms and Computational Thinking

Autumn 2016

Thursday, 6th October 2016

Exercise 3 - Programming basics

This exercise will first present the preliminary information to create a very simple program in Python, Scala and Swift and, based on it, you will be able to do the exercises of the second part.

1. Preliminary information

Here you will find some useful information helping to complete the exercises of the second section.

How to declare a variable?

Python : *variableName* = *variableValue*

Scala : *var variableName* = *variableValue* OR *var variableName* : *variableType* = *variableValue*

Swift : *var variableName* = *variableValue*

The *variableValue* can be a string of characters, an integer, a float, a boolean... In the case of a string, you must add " around the value of the variable. If the variable is a number, you can directly write the value.

How to create a constant?

Python : no possibility to create a constant!

Scala : *val constantName* = *constantValue*

Swift : *let constantName* = *constantValue*

How to get and store a value from a user (-> input)?

Python : *variableName* = input('text')

Scala : *var variableName* = scala.io.StdIn.readLine()

Swift : `var variableName = readLine(strippingNewline : true)!`

When you get a value from a user, this value is a string. You must find a way to convert this value into an integer or another number type if you need a number.

How to print a variable or a constant (-> output) ?

Python : `print(variableName)`

Scala : `print(variableName)`

Swift : `print(variableName)`

How to create a conditional block ?

Python :

```
if condition1 :  
    ...  
elif condition2 :  
    ...  
else :  
    ...
```

Scala :

```
if (condition1) {  
    ...  
} else if (condition2) {  
    ...  
} else {  
    ...  
}
```

Swift :

```
if condition1 {  
    ...  
} else if condition2 {  
    ...  
} else {  
    ...  
}
```

In order to create a condition, you need to use comparison operators with variables, such as *equal to* is `==`, *not equal to* is `!=`, *greater than* is `>`, *greater than or equal to* is `>=`, etc.

2. Exercises

For each exercise, you must create a Python, a Scala and a Swift script and be able to run them with command line. To begin, create a new folder, called *Exercise3* for example, on the Desktop.

How to create and run a script via the terminal ?

Python : Create a new file and its content by using *Atom* with the extension *.py* that you will put in the folder. Open a terminal window and write the following command line (if you do not have python version 3, you must download it with *Homebrew* with this command line : `brew install python3.5`) : `python3.5 filePath`

Scala : Create a new file and its content by using *Atom* with the extension *.scala* that you will put in the folder. Open a terminal window and change the location of command line execution with this command line : `cd folderPath`. You must first compile your code with this command line : `scalac filePath`. Then, you can execute your code with this command line : `scala -classpath . className`.

Swift : Create a new file and its content by using *Atom* with the extension *.swift* that you will put in the folder. Open a terminal window and change the location of command line execution with this command line : `cd folderPath`. You must first compile your code with this command line : `swiftc filePath`. Then, you can execute your code with this command line : `./fileName`.

With all information given above, you should be able to do the following exercises below :

1. Create a script that displays if a given number (ask the user to give a number (input) and store it into a variable) is positive, negative or equal to 0.
2. Create a script that displays if a given number (ask the user to give a number (input) and store it into a variable) is even or odd (the modulo operator is `%`).
3. Create a script in order to do a monetary calculation with two variables, the first variable is the money given to pay a bill and the second variable is obviously the bill. Try the following values : `1.03 / 0.42` and `1.00 / 0.9` (money / bill). Display the amount of change by subtracting the bill from the given money. Do you obtain a correct and reliable result ? If you are not able to find a correct result, try to find a way to obtain it by using a specific type (use a search engine such as Google, DuckDuckGo...).